THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 59, No. 10, December 1980
Printed in U.S.A.

Analysis of Decision-Directed Equalizer Convergence

By J. E. MAZO

(Manuscript received July 21, 1980)

Digital data signals are usually equalized by passing samples of the received signal through an adaptive equalizer consisting of a tapped delay line having adjustable coefficients (tap weights). The equalizer tap weights are adjusted by starting the transmission with a short training sequence of digital data known in advance by the receiver. This paper analyzes the situation when the known training sequence is replaced by a sequence of data symbols estimated from the equalizer output and treated as known data. Such procedures are called "decision-directed" startup. With a known training sequence, the "least-mean-square" adjustment algorithm corresponds mathematically to searching for the unique minimum of a quadratic "error" surface whose unimodal nature assures convergence. In decision-directed startup, by contrast, the use of estimated and unreliable data changes the error surface into a multimodal one so that complex behavior may result. We describe the nature of the error surfaces for binary and four-level transmission, thereby gaining insight into convergence problems. The most significant conclusion is that a poor choice for the initial tap settings may result in the taps converging to an undesirable setting. We show that, because of finite step-size effects, fluctuations are significant at the undesired settings and cause the spurious capture to have a long, but finite, duration. Finally we provide information on stability, convergence times, and lifetimes and their relation to the adaptation parameter (step size).

I. INTRODUCTION

In high-speed data transmission (4.8 or 9.6 kilobit/s) over voice-grade telephone channels, it is necessary to compensate for the linear amplitude and phase distortion to which the data signal will be subjected. This compensation is usually accomplished by passing samples of the received signal through an adaptive equalizer consisting of a tapped delay line having adjustable coefficients (tap weights).

Since the distortion is initially unknown, the tap weights must be suitably adjusted. Conventionally, the equalizer tap weights are adapted by starting the transmission with a short training sequence of digital data known in advance by the receiver. The receiver then uses the difference between the equalizer output signal and the known data to adjust tbe tap weights.

In modern data-communication environments the above method may not always be practical, and thus new training procedures which do not make use of a known data-training sequence are required.

A natural suggestion is to replace the known training sequence with a sequence of data symbols estimated from the equalizer output, and treat these as if they were known data. Such procedures are often called "decision-directed" startup. However, when these decision-directed startup procedures are used the estimated data may be unreliable, so that it is not even certain that the tap weights will converge to their correct settings.

For example, assume that there are N tap weigbts, $c_1, c_2, \cdots, c_N$, to be adjusted. The collection of these numbers is to be regarded as a vector $\mathbf{c}$ in an abstract N-dimensional space. For the case of a known training sequence, the conventional tap-adjustment algorithm for finding the optimum tap settings (called the least-mean-square algorithm) corresponds mathematically to searching for the unique minimum of a certain quadratic "error" surface defined in this $\mathbf{c}$ space. The simple unimodel nature of this surface assures convergence. In decision-directed startup, by contrast, the use of estimated and unreliable data changes the error surface being searched into a multimodal one, so that quite complex behavior may result. The local minima are of two types. First there are the desired local minima, ones whose positions correspond to tap settings yielding the same performance as if known data were used. Second, there are the undesired, or extraneous, local minima which appear at positions corresponding to tap settings yielding inferior equalizer performance.

We begin our work in Section III by describing the nature of the decision-directed error surfaces for binary (± 1) and four-level ($\pm 1, \pm 3$) transmission. In general, the surfaces in N dimensions are too complex for an exact description to be given. However, low-dimensional examples give considerable insight into the problems encountered with convergence. Our most significant conclusion is that a poor choice for the initial tap settings may result in the taps converging to an undesirable setting and remaining there for a long time. In Section IV we show that although random fluctuations of the taps about the desired minima are small, this is not the case for random fluctuations about the extraneous minima. Rather, being trapped at one of the extraneous minima is an event having a long but finite lifetime. These lifetimes depend on the geometry of the error surface, and on the adaptation

parameter (step size) of the algorithm. Finally, we give quantitative information on stability, convergence times, and lifetimes, and their relation to the step size, although it is sometimes necessary to resort to approximations and idealized geometries to do so, even for the simplified mathematical model that we consider.

Gitlin and Werner[1] have made an experimental study of decision-directed startup. They discovered that using the least-mean-square algorithm to update the tap weights works with estimated data in the binary case, but not in the four-level case. Other surprising phenomena have been observed. For example, E. Y. Ho[2] observed that occasionally, with four-level data, the signal constellation at the equalizer output would be perfect, except for one item. The signal points would be so reduced in amplitude that all data would be decoded as ± 1, yielding an error rate of one-half. No ready explanation for these observations was at hand. Subsequent prodding from J. Salz lead us to conduct the present investigation, and, in the course of our general study, explanations of the above phenomena were found.

II. MODEL AND REVIEW

We begin this section with a description of the model that we use. As with many mathematical investigations, we have a choice as to what should be included in the formulation of the problem. Here, although we make several simplifications for mathematical tractability, our simplified model will provide an understanding of the unusual phenomena that have been observed in the experimental studies of decision-directed startup.

The model is as follows: We consider baseband transmission of independent, equiprobable, binary or four-level data over a noiseless channel with no distortion. The receiver is an N-tap synchronous equalizer whose initial tap setting is assumed arbitrary. We study the subsequent convergence of the tap vector to a final value when the mean-square-adjustment algorithms are modified in an obvious way to include the case of estimated data. The equations appropriate to the model are given at the start of Section III.

As to notation, the vector of real tap coefficients is denoted by $\mathbf{c} = (c_1, \cdots, c_N)$.† The kth data symbol is denoted by a_k, while at time n, $n = 0, 1, 2, \cdots$, the equalizer output y_n is, for any fixed tap setting $\mathbf{c}$,

$$y_n = \mathbf{c} \cdot \mathbf{a}_n, \tag{1}$$

where

$$\mathbf{a}_n = (a_n, a_{n+1}, \cdots, a_{n+N-1}) \tag{2}$$

† In the mathematics, all vectors are column vectors. In sentences, or in listing, the vector is, for typographic convenience, written as a row without using the usual superscript plus (+) to denote transposition.

is the vector of N channel samples that would be stored in the equalizer for this ideal situation.

Ideally, we desire the output sequence (1) to be the sequence of data symbols. For this to occur, an ideal tap vector might be, for example, $\mathbf{c} = (1, 0, 0, \cdots, 0)$ or, in fact, any vector $\mathbf{c}$ having exactly one component unity and the rest zero. For each such choice of tap vector the sequence of data values is reproduced, but with a different (and unimportant) time delay. For the present problem the set of desirable tap vectors must be enlarged to include the negatives of those just described, as well. The data must then be differentially encoded.

We conclude this section with a review of the least-mean-square (LMS) algorithm, and its analysis, for the case when the data are known by the equalizer. This review has two purposes. First it prepares the way for similar considerations in blind startup, and also it introduces some approximations that will be used throughout the paper.

With known data, the optimum tap vector is defined to be the vector, $\mathbf{c}_{\text{opt}}$, which minimizes the mean-square error $\mathscr{E}^2$, where

$$\mathscr{E}^2 = E[\mathbf{c}\cdot\mathbf{a}_n - a_l]^2, \tag{3}$$

E denoting expectation with respect to all data symbols, and $\mathbf{c}\cdot\mathbf{a}$ denoting the inner product between the vectors $\mathbf{c}$ and $\mathbf{a}$. Of course, $\mathbf{c}\cdot\mathbf{a} = c^+a$. Regarding (3), note that (2) restricts l to satisfy $n \le l \le n + N - 1$ so that a meaningful problem will result. For definiteness we choose $l = n$ so that (3) becomes

$$\mathscr{E}^2 = E[\mathbf{c}\cdot\mathbf{a}_n - a_n]^2, \tag{4}$$

where, again, the expectation is with respect to all the data symbols $\{a_n\}$, and $\mathbf{c}$ is a generic point in tap space. Regarded as a function of $\mathbf{c}$, the right member of (4) describes the mean-square-error surface.

We observe that the data symbol a_n satisfies

$$Ea_n = 0, \tag{5}$$

$$Ea_n^2 \equiv \sigma_a^2 = \begin{cases} 1 \text{ binary}, \\ 5 \text{ four-level}, \end{cases} \tag{6}$$

while the data vector $\mathbf{a}_n$ satisfies, using independence of the data symbols,

$$E\mathbf{a}_n\mathbf{a}_n^+ = \sigma_a^2 I. \tag{7}$$

In (7), I is the identity matrix for N dimensions. It is also customary to denote $Ea_n\mathbf{a}_n$ by $\sigma_a^2\mathbf{v}$, $\mathbf{v} = (1, 0, 0, \cdots, 0)$. Then, from (4), the error surface $\mathscr{E}^2$ may be written

$$\mathscr{E}^2 = \sigma_a^2[1 + \mathbf{c}^+\mathbf{c} - 2\mathbf{c}^+\mathbf{v}]. \tag{8}$$

This surface is convex and has a unique minimum at $\mathbf{c} = \mathbf{c}_{opt} = \mathbf{v}$. The value $\mathscr{E}_0^2$ of $\mathscr{E}^2$ at the minimum is

$$\mathscr{E}_0^2 = 0. \tag{9}$$

In applications, the error surface is unknown, and an iterative gradient search, called the LMS algorithm, is used to find $\mathbf{c}_{opt}$. If at the nth iteration the tap vector had the value $\mathbf{c}_n$, the known-data algorithm for our model is

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \alpha e_n \mathbf{a}_n, \tag{10}$$

where

$$e_n = \mathbf{c}_n \cdot \mathbf{a}_n - a_n \tag{11}$$

is the instantaneous output error. The step-size parameter α determines the stability and speed of convergence. To see this explicitly, denote the error vector after the nth iteration, $\mathbf{c}_n - \mathbf{c}_{opt}$, by $\boldsymbol{\epsilon}_n$. Then it can be shown that (10) and (11) give

$$\boldsymbol{\epsilon}_{n+1} = (I - \alpha \mathbf{a}_n \mathbf{a}_n^+) \boldsymbol{\epsilon}_n, \tag{12}$$

and so

$$E \| \boldsymbol{\epsilon}_{n+1} \|^2 = E\, \boldsymbol{\epsilon}_n^+ (I - \alpha \mathbf{a}_n \mathbf{a}_n^+)^2 \boldsymbol{\epsilon}_n. \tag{13}$$

To evaluate the expectation in the right member of (13), it is standard practice to assume $\boldsymbol{\epsilon}_n$ and $\mathbf{a}_n$ are statistically independent. This assumption works surprisingly well in practice, and here and henceforth in our paper this so-called "independence assumption" is made. More innocently, because it may be checked by exact calculation, we approximate†

$$\begin{aligned} E(\mathbf{a}_n \mathbf{a}_n^+)(\mathbf{a}_n \mathbf{a}_n^+) &= E[(\mathbf{a}_n \cdot \mathbf{a}_n)\mathbf{a}_n \mathbf{a}_n^+] \\ &\approx E(\mathbf{a}_n \cdot \mathbf{a}_n) E(\mathbf{a}_n \mathbf{a}_n^+) = N\sigma_a^4 I. \end{aligned} \tag{14}$$

Thus (13) becomes, on taking the expectation,

$$E \| \boldsymbol{\epsilon}_{n+1} \|^2 = (1 - 2\alpha\sigma_a^2 + N\sigma_a^4 \alpha^2) E \| \boldsymbol{\epsilon}_n \|^2. \tag{15}$$

From (15) we see that the algorithm converges if $(1 - 2\alpha\sigma_a^2 + N\sigma_a^4\alpha^2) < 1$ or, in other words, if

$$0 < \alpha < \frac{2}{N\sigma_a^2}. \tag{16}$$

† We have exactly that $E[(\mathbf{a}_n \cdot \mathbf{a}_n)\mathbf{a}_n \mathbf{a}_n^+] = N\sigma_a^4 I + [Ea_n^4 - \sigma_a^4]I$. Now $[Ea_n^4 - \sigma_a^4] = 0$ for the binary case and equals -4.5 for the four-level case. It may be neglected with respect to the first term even when N, the number of taps, is only moderately large.

Convergence is most rapid when $(1 - 2\alpha\sigma_a^2 + N\sigma_a^4\alpha^2)$ is smallest, i.e., when

$$\alpha = \alpha_0 = \frac{1}{N\sigma_a^2}. \tag{17}$$

We close this section with some more notation. During transmission the data symbols are determined by "slicing" the equalizer output y_n. We name this nonlinear function $\mathrm{sl}(\cdot)$, for slicer. It is defined for binary transmission by

$$\mathrm{sl}(x) = \mathrm{sgn}(x) \qquad \text{(binary)}$$

and for four-level transmission by

$$\mathrm{sl}(x) = -\,\mathrm{sl}(-x) = \begin{cases} 1 & \text{if} \quad 0 < x < 2, \\ 3 & \text{if} \quad x > 2. \end{cases} \qquad \text{(4-level)}$$

III. DECISION-DIRECTED SURFACES

The standard modification of (10) and (11), which is appropriate to decision-directed startup, and which we analyze in this work, is simple to describe. Instead of (10) and (11) we have

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \alpha\hat{e}_n\mathbf{a}_n, \tag{18}$$

$$\hat{e}_n = \mathbf{c}_n \cdot \mathbf{a}_n - \hat{a}_n, \tag{19}$$

where

$$\hat{a}_n = \mathrm{sl}(\mathbf{c}_n \cdot \mathbf{a}_n). \tag{20}$$

Thus we have replaced the known-data symbol a_n in (11) by its estimate (20).†

The task undertaken in this section is to describe the error surface that goes along with (18)–(20). That is, we want to give the equivalents of (4) and (8) which apply for known data. Since $\hat{e}_n$ serves as an estimated gradient in (18), the surface, which we call $\mathscr{F}^2$, is, in principle, described by

$$\mathscr{F}^2 = E[\mathbf{c}\cdot\mathbf{a}_n - \hat{a}_n]^2 = E[\mathbf{c}\cdot\mathbf{a}_n - \mathrm{sl}(\mathbf{c}\cdot\mathbf{a}_n)]^2, \tag{21}$$

$\mathbf{c}$ being a generic point in tap space. Averaging over the data vector $\mathbf{a}_n$ imposes the major difficulty. By stationarity, the average in (21)

† To one unfamiliar with the actual data-transmission algorithm, it no doubt seems absurd to regard $\mathbf{a}_n$ as known, as in (18), but yet a_n, its first component, is not known. Actually in the real problem, $\mathbf{a}_n$ in (18)–(20) is replaced by a vector which is measured. For the ideal channel the measured value would, in fact, be $\mathbf{a}_n$, but the point is that the machine which implements the algorithm is built to handle the general case and would not know of, nor could it make use of, this fact. A similar remark could have been made in the treatment of (10) and (11).

doesn't depend on n and in such situations we often drop the subscript, writing simply $\mathbf{a}$ for $\mathbf{a}_n$. The components of $\mathbf{a}$ are then $(a_1, \cdots, a_N)$. Having mentioned this, we trust that no confusion will arise with the convention established in (2).

We begin with the binary case. Applying (6) and (7) to (21) gives

$$\mathscr{F}^2 = 1 + \mathbf{c}\cdot\mathbf{c} - 2E\,|\,\mathbf{c}\cdot\mathbf{a}\,|, \tag{22}$$

where the last term in (22) must still be averaged over the 2^N binary vectors $\mathbf{a}^{(i)}$, $i = 1, 2, \cdots, 2^N$. Now note that, for a fixed i, the hyperplane $\mathbf{c}\cdot\mathbf{a}^{(i)} = 0$ divides N-space into two regions, depending on the sign of $\mathbf{c}\cdot\mathbf{a}^{(i)}$; in one region, $\mathbf{c}\cdot\mathbf{a}^{(i)} > 0$, while $\mathbf{c}\cdot\mathbf{a}^{(i)} < 0$ in the other. Then the entire collection of such hyperplanes divides N-space into a number of cone-shaped regions with the property that, in each cone, $\mathbf{c}\cdot\mathbf{a}^{(i)}$ has a fixed sign (which depends on i). Suppose then, that $\mathbf{c}$ is in one of these cones, called $\mathscr{R}$, and let $\mathscr{S}$ denote the set of indices $\{i\}$ for which $\mathbf{c}\cdot\mathbf{a}^{(i)} > 0$, so $|\,\mathbf{c}\cdot\mathbf{a}^{(i)}\,| = \mathbf{c}\cdot\mathbf{a}^{(i)}$, $i \in \mathscr{S}$. Denote the complement set of indices by $\mathscr{S}^c$. With this notation (22) becomes, for $\mathbf{c} \in \mathscr{R}$,

$$\begin{aligned}\mathscr{F}^2 &= 1 + \mathbf{c}\cdot\mathbf{c} - \frac{2}{2^N}\left[\sum_{i\in\mathscr{S}} \mathbf{c}\cdot\mathbf{a}^{(i)} - \sum_{i\in\mathscr{S}^c} \mathbf{c}\cdot\mathbf{a}^{(i)}\right] \\ &= 1 + \mathbf{c}\cdot\mathbf{c} - 2\mathbf{c}\cdot\frac{1}{2^N}\left[\left(\sum_{i\in\mathscr{S}} \mathbf{a}^{(i)} - \sum_{i\in\mathscr{S}^c} \mathbf{a}^{(i)}\right)\right] \\ &= 1 + \mathbf{c}\cdot\mathbf{c} - 2\mathbf{c}\cdot\mathbf{c}_0, \end{aligned} \tag{23}$$

where $\mathbf{c}_0$ is defined by (23) in the obvious way. Since the quadratic form in (23) is strictly positive definite, the function $\mathscr{F}^2$ has a unique minimum in the region $\mathscr{R}$, at $\mathbf{c} = \mathbf{c}_0$, provided that the vector $\mathbf{c}_0 \in \mathscr{R}$. If $\mathbf{c}_0 \notin \mathscr{R}$, $\mathscr{F}$ is convex but has no minimum interior to $\mathscr{R}$. In the former case, we denote the value of $\mathscr{F}^2$ at the minimum by $\mathscr{F}_0^2$, and we have

$$\mathscr{F}_0^2 = 1 - \mathbf{c}_0\cdot\mathbf{c}_0 > 0. \tag{24}$$

The above discussion shows that $\mathscr{F}^2$ always has its quadratic part, $\mathbf{c}\cdot\mathbf{c} = \mathbf{c}^+I\mathbf{c}$, determined by the identity matrix, while the linear term, $-2\mathbf{c}\cdot\mathbf{c}_0$, changes from region to region; we expect a different $\mathbf{c}_0$ for each region. But since $|\,x\,|$ is a continuous function, we see, from (22), that $\mathscr{F}^2$ is also continuous.

Counting the number of cone-shaped regions appears to be very difficult in general. The problem is equivalent to the following. Let 0, the origin, be at the center of an N-cube, and consider all hyperplanes through 0 which are perpendicular to some vertex vector. Into how many cones do these hyperplanes divide N-space? For $N = 2, 3, 4, 5,$

there are 4, 14, 104, 1882 cones, respectively.† We obtain sufficient insight for our purposes by considering some low-dimensional examples of (23).

For $N = 2$, the lines through the origin which are perpendicular to the vertex vectors divide the plane into four regions, as shown in Fig. 1. Calculating $\mathbf{c}_0$ for region I, for example, gives, using (23),

$$\mathbf{c}_0 = \frac{1}{4}\left[\begin{pmatrix}1\\1\end{pmatrix} + \begin{pmatrix}1\\-1\end{pmatrix} - \begin{pmatrix}-1\\1\end{pmatrix} - \begin{pmatrix}-1\\-1\end{pmatrix}\right] = \begin{pmatrix}1\\0\end{pmatrix}. \tag{25}$$

Its position along the c_1 axis is indicated by a small circle (as are the $\mathbf{c}_0$ vectors for the other regions). Since $\mathbf{c}_0$ is actually in region I, we have a local minimum there with $\mathscr{F}_0^2 = 0$. Note that if a data vector $\mathbf{a} = (a_1, a_2)$ is sent, then $\mathbf{c}\cdot\mathbf{a} = \mathbf{c}_0\cdot\mathbf{a} = a_1$, and perfect detection of the first symbol occurs. Note $\mathscr{E}^2$ in (8) also has its unique minimum at this point. However, from (19) and (20), $\hat{e}_n$ will always be zero if $\mathbf{c} = \mathbf{c}_0 = (-1, 0)$ as well. Thus $\mathscr{F}^2$ has a local minimum there too, and also at the points $\mathbf{c}_0 = (0, \pm 1)$ (the second symbol is also a valid one to use for detection). Such symmetries will always occur, and we consider just a representative $\mathbf{c}_0$, e.g., $(1, 0, 0, \cdots, 0)$, for general N. This vector is representative of one of $2N$ positions to which we would wish the algorithm to converge. For $N = 2$, no other minima occur.

The case $N = 3$ is the first interesting one. To describe the cones, we have shown their intersection with the cube in Fig. 2. Two types of cones occur. There are four-sided ones which intersect the faces in squares, and three-sided ones having their axes along the vertex directions. Representative minima of $\mathscr{F}^2$ occur at $\mathbf{c}_0 = (1, 0, 0)$ and $\mathbf{c}_0 = (1/2)(1, 1, 1)$. Thus there are six minima of the first type (centers of faces), and eight of the second type (along vertex directions). At the former, $\mathscr{F}_0^2 = 0$; at the latter, $\mathscr{F}_0^2 = 0.25$.

If we are in the region containing $(1, 0, 0)$, we always make a correct decision (on the first symbol of the $\mathbf{a}$ vector, it turns out) and the probability of error $P_e = 0$. If we are in the region containing $(1, 1, 1)$, the data vectors $\mathbf{a} = (1, 1, 1)$, $(1, 1, -1)$, $(1, -1, 1)$, and their negatives always have their first symbol decoded correctly [i.e., $a_1 = \text{sgn}(c\cdot a)$] whereas $(1, -1, -1)$ and its negative give an incorrect value. Thus $P_e = 1/4$ for the first symbol when the tap vector is in this region. For this situation it happens that $P_e = 1/4$ for any other symbol too.

Thus, for $N = 3$ we see that if we choose a bad initial state for the equalizer, namely an initial tap vector lying in a vertex cone, convergence via gradient search will be to the local minimum at $(1/2)(1, 1, 1)$. For all practical purposes, it will, because of initial conditions, have converged during decision-directed startup to an undesired set of tap

† A list of the number of regions for N up to ten is given in Ref. 3.

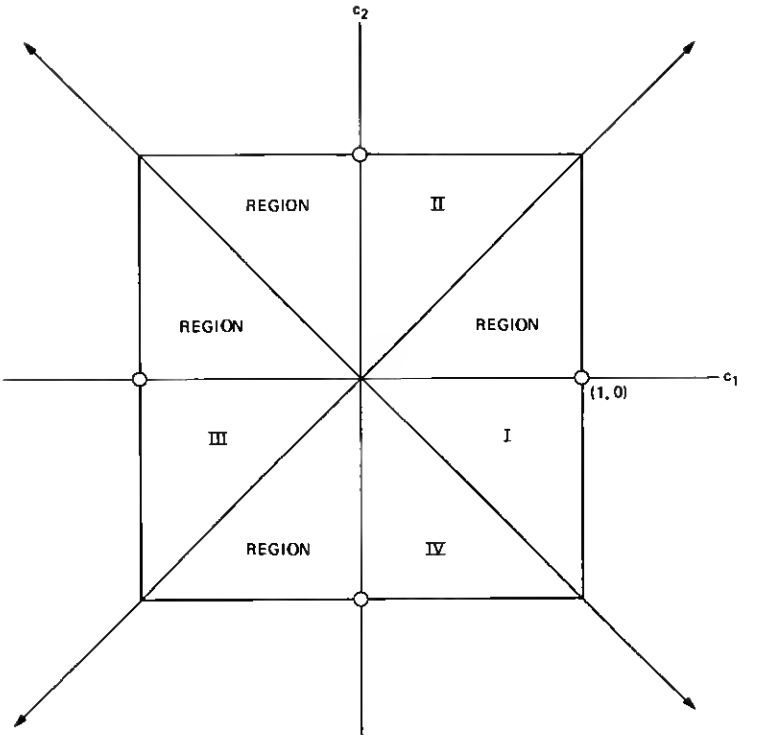

Fig. 1—Cones for $N = 2$.

weights. We are assuming here that the LMS algorithm behaves as if the true gradient of the surface were being used. This is essentially true if the step size α is small enough; more comments on this will be made later.

Similar situations prevail in five dimensions, where local minima occur when the taps are proportional to the representative vectors (10000), (11100), (11111), (53311), and (22111). There are also two other classifications of cones which do not have local minima in their interiors.

The situation which includes noise and distortion should be clear. Certain unknown optimum tap settings exist, one of which we would hope to converge to, during decision-directed startup. If we make an initial guess close to such a desired local minimum, we converge there. If not, we converge to an undesired setting, yielding a bad error rate. Later, when we consider fluctuations for a finite step size, we shall see that capture at a spurious minimum need not be permanent; capture at a desired local minimum will be.

At this point we stop our investigation of the binary problem and

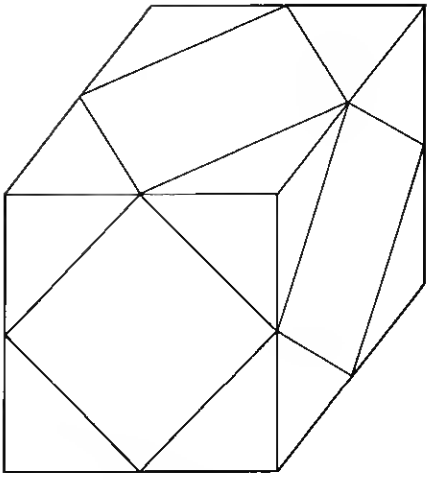

Fig. 2—Cones for $N = 3$.

move briefly to the four-level case. For this new situation we have, in place of (22), the representation

$$\mathscr{F}^2 = \sigma_a^2 \mathbf{c}\cdot\mathbf{c} + E[\mathrm{sl}^2(\mathbf{c}\cdot\mathrm{a}) - 2\mathbf{c}\cdot\mathbf{a}\ \mathrm{sl}(\mathbf{c}\cdot\mathbf{a})], \tag{26}$$

where $\mathrm{sl}^2(x)$ means $(\mathrm{sl}(x))^2$. In (26) the average is taken with respect to all 4^N equilikely vectors $\mathbf{a}$, which have ± 1, ± 3 as components. Note that $\mathscr{F}^2$ in (26) is a continuous function of $\mathbf{c}$, because $\mathrm{sl}^2(x) - 2x\ \mathrm{sl}(x)$ is continuous.

We again partition N-space into regions, where now in each region $\mathrm{sl}(\mathbf{c}\cdot\mathbf{a}^{(i)})$ is constant for each fixed i, $i = 1, 2, \cdots, 4^N$. The averaging indicated in (26) again leads to a quadratic-plus-linear structure within each region, although the regional map is now considerably more complex that in the binary case. Its most outstanding feature is that the regions are now not cone-shaped. A map of regions for the four-level, $N = 2$ problem is drawn in Fig. 3 where the additional complexity is readily apparent. The error-free regions about the optimum tap vectors are indicated by the small kitelike regions, cross hatched in the figure. A much more accurate guess would have to be made with four-level transmission to assure that one had error-free data in a decision-directed startup procedure.

For the four-level case in N dimensions we still have local (and global) minima at the optimum tap values represented by (1, 0, 0, $\cdots$, 0), $\mathscr{F}_0^2 = 0$, and other local minima as well. Although we have made no attempt to describe all the other local minima, there is one class that we do mention. We find it by looking for a local minimum of (26) at $\mathbf{c} = \mathbf{c}_0 = (g, 0, 0, \cdots, 0)$, $g > 0$. In the neighborhood of such a

tap vector we have $\mathrm{sl}(\mathbf{c}\cdot\mathbf{a}) = \mathrm{sl}(g a_1)$. If $0 < g < 2/3$, then $\mathrm{sl}(g a_1) = \mathrm{sgn}(a_1)$, and we have

$$\mathscr{F}^2 = \sigma_a^2\, \mathbf{c}\cdot\mathbf{c} + 1 - 2E\, \mathbf{c}\cdot\mathbf{a}\, \mathrm{sgn}\, a_1 = 1 + \sigma_a^2 \mathbf{c}\cdot\mathbf{c} - 4c_1. \tag{27}$$

It follows from (27) that there is a minimum at

$$\mathbf{c} = (\tfrac{2}{5}, 0, 0, \cdots, 0), \tag{28}$$

which

$$\mathscr{F}_0^2 = 0.2. \tag{29}$$

A graph of $\mathscr{F}^2$, as we move out along the $\mathbf{c}_1$ axis, is independent of

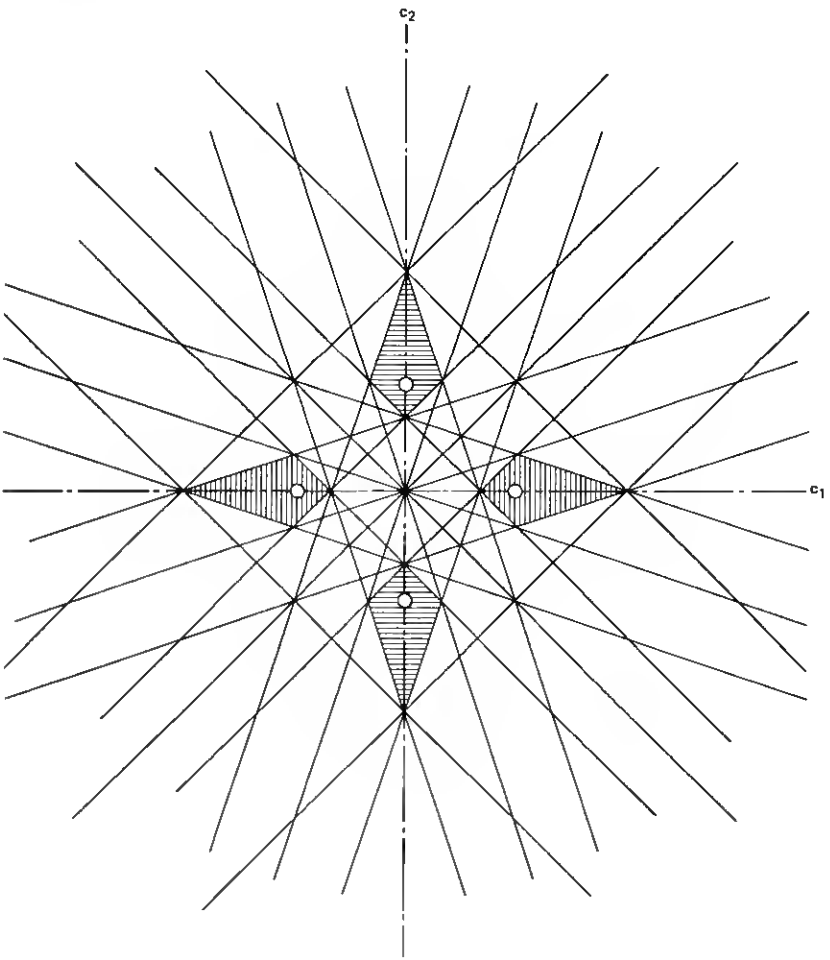

Fig. 3—Regions for $N = 2$, four-level transmission.

the dimension N, and is shown in Fig. 4. In particular, the minimum at the value of $\mathbf{c}$ given by (28), and also the global optimum, are to be noted.

The character of the equalizer output when the tap vector is trapped at (28) may be noted. Instead of observing the ± 1, ± 3 data values we would see $\pm 2/5$, $\pm 6/5$, all of which would be decoded as ± 1. E. Y. Ho[2] has, in fact, observed such contracted signal constellations during startup experiments.

An approximate description of a large number of other minima is deferred to Appendix A.

IV. FINITE STEP SIZE

In Section III we described the error surface appropriate to decision-directed startup with the mean-square algorithm and showed that it had many minima. Further, we assumed for sufficiently small step size α that the local motion on this surface followed the gradient directions.

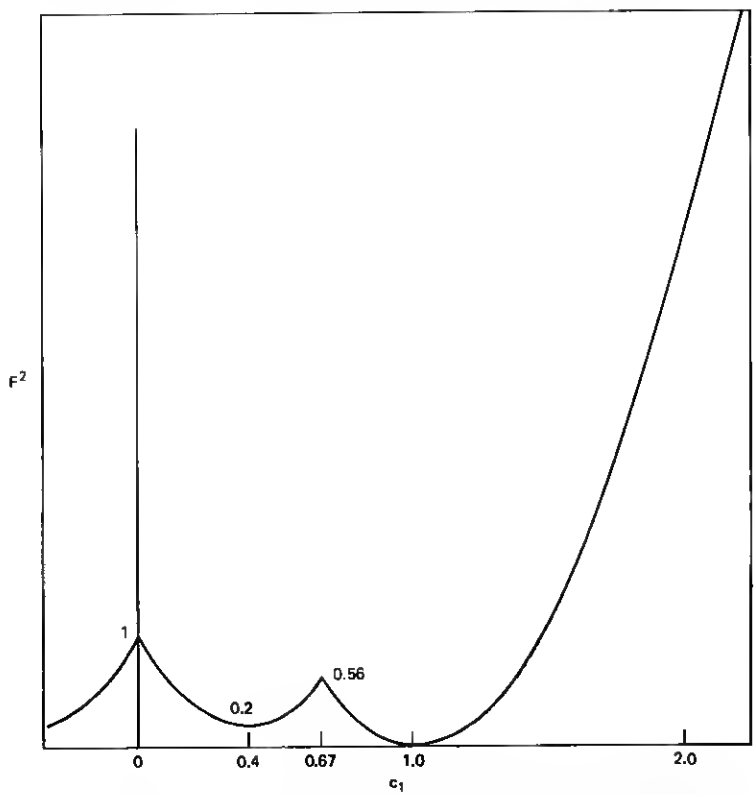

Fig. 4—The surface $\mathscr{F}^2$ along the c_1 axis (four-level transmission).

This implies that we would reach equilibrium at a local minimum and remain there. For finite step size this is a useful picture, but it is only an approximate one. The most important correction that we must make to it is to realize that the extraneous minima of $\mathscr{F}^2$ that we have discovered are not truly stable. We will, if only we wait long enough, always reach one of the global minima with $\mathscr{F}^2 = 0$.

To illustrate this, assume that we are initially in a region possessing a local minimum at $\mathbf{c} = \mathbf{c}_0$, and that we remain in that region for a long time. Then one may derive an equation for the behavior of the mean norm of the error vector $\boldsymbol{\epsilon}_n = \mathbf{c}_n - \mathbf{c}_0$. In fact, subtracting $\mathbf{c}_0$ from both sides of (18) gives

$$\boldsymbol{\epsilon}_{n+1} = \boldsymbol{\epsilon}_n - \alpha \mathbf{a}_n[\mathbf{a}_n \cdot \boldsymbol{\epsilon}_n + \mathbf{c}_0 \cdot \mathbf{a}_n - \mathrm{sl}(\mathbf{c}_n \cdot \mathbf{a}_n)]. \tag{30}$$

By definition of our regions and the assumption that we do not leave the region, we have

$$\mathrm{sl}(\mathbf{c}_n \cdot \mathbf{a}_n) = \mathrm{sl}(\mathbf{c}_0 \cdot \mathbf{a}_n), \qquad \text{all } n. \tag{31}$$

Letting

$$Q_0 = \mathbf{c}_0 \cdot \mathbf{a}_n - \mathrm{sl}(\mathbf{c}_0 \cdot \mathbf{a}_n), \tag{32}$$

and squaring both members of (30) and taking averages with respect to all data symbols, we have

$$E\boldsymbol{\epsilon}_{n+1}^2 = E[\boldsymbol{\epsilon}_n^2 - 2\alpha(\boldsymbol{\epsilon}_n \ \mathbf{a}_n)^2 - 2\alpha Q_0 \boldsymbol{\epsilon}_n \cdot \mathbf{a}_n] + \alpha^2 E\mathbf{a}_n^+ \mathbf{a}_n[(\boldsymbol{\epsilon}_n \cdot \mathbf{a}_n)^2 + 2Q_0 \boldsymbol{\epsilon}_n \cdot \mathbf{a}_n + Q_0^2], \tag{33}$$

where, for notational simplicity, we have set $\boldsymbol{\epsilon}_n^2 = \| \boldsymbol{\epsilon}_n \|^2$. Then it may be shown, from (33), using approximations of the type described in Section II, that we have, approximately,

$$E\boldsymbol{\epsilon}_{n+1}^2 = (1 - 2\alpha\sigma_a^2 + N\alpha^2\sigma_a^4)E\boldsymbol{\epsilon}_n^2 + N\alpha^2\sigma_a^2\mathscr{F}_0^2. \tag{34}$$

Thus, from (34), as n becomes large the average of the squared-error vector approaches

$$E\boldsymbol{\epsilon}_\infty^2 = \frac{\alpha N}{2 - \alpha N\sigma_a^2}\mathscr{F}_0^2. \tag{35}$$

To insure rapid initial convergence one normally chooses $\alpha N\sigma_a^2 = 1$, and for this choice of α, (35) becomes $E\boldsymbol{\epsilon}_\infty^2 = \mathscr{F}_0^2/\sigma_a^2$.

Stability requires $\alpha N\sigma_a^2 < 2$, as is readily apparent from (34).

If we are in equilibrium about $\mathbf{c}_{\mathrm{opt}}$, then $\mathscr{F}_0^2 = 0$ and, from (35), there are no fluctuations. However, consider the binary case with $N = 3$, $\mathbf{c}_0 = (1/2)(1, 1, 1)$, and $\alpha = 1/N$. For this case, $E\boldsymbol{\epsilon}_\infty^2 = \mathscr{F}_0^2 = 0.25$; thus a typical error vector might have length about $\sqrt{E\boldsymbol{\epsilon}_\infty^2} = \sqrt{0.25} = 0.5$. But the distance from $\mathbf{c}_0$ to (1, 0, 0) is only $\sqrt{0.75} = 0.87$. Certainly it is reasonable to expect that fluctuations would soon move $\mathbf{c}$ from the

region containing $\mathbf{c}_0$ to the error-free region containing $\mathbf{c}_{opt} = (1, 0, 0)$, with convergence to $\mathbf{c}_{opt}$ resulting.

As we note from (35), the mean-squared fluctuation decreases for small α. Thus, for α small, we expect to wait a very long time for deviations of the required magnitude to occur, and our earlier assumption of being trapped at an undesired minimum is, in this sense, justified.

Examining the detailed mechanism causing ultimate convergence to a $\mathbf{c}_{opt}$ for the above example is worthwhile. For definiteness, consider convergence to (1, 0, 0). Table I illustrates the possible $\mathbf{a}$ vectors (only four of the eight need be listed) and the resulting decisions on the first symbol.

In any infinitely-long time sequence of independently chosen vectors $\mathbf{a}$, there will occur, if we wait, long runs where the vector $(1, -1, -1)$ does not occur. Then we have no errors in the first symbol, and the tap vector moves, if the run is long enough, to a neighborhood of (1, 0, 0), after which no errors occur, independently of what the succeeding $\mathbf{a}$ vectors are. In this manner we can imagine, in higher-dimensional problems, special sequences, low in errors for the kth symbol, causing the tap vector to move from region to another region, until the error-free region about the kth coordinate axis is entered.

For small step size a diffusion approximation should describe the randomness quite well. However, the difficulty that we have in describing (or even counting) the regions in N-dimensions prevents such an approach from giving precise information as to convergence times. Nevertheless some model problems are considered in Appendix B.

Simulations show that, for the binary problem, some moderate delay is experienced with regard to convergence to $\mathbf{c}_{opt}$ when starting as a random position with $\alpha = 1/N$. The delay does become excessive for four-level transmission. This may be due to the smaller error-free region which must be reached.

APPENDIX A

Approximate Description of Some Minima

The discussion in Section III emphasized the great plurality of regions and local minima associated with the surface represented by

Table I—Decision table for $\mathbf{c}_0 = \frac{1}{2}(1, 1, 1)$

$\mathbf{a}$	$\mathbf{c}_0 \cdot \mathbf{a}$	$\hat{a} = \text{sgn}(\mathbf{c}_0 \cdot \mathbf{a})$
(1, 1, 1)	3	1 (correct)
(1, 1, −1)	1	1 (correct)
(1, −1, 1)	1	1 (correct)
(1, −1, −1)	−1	−1 (error)

(21). A natural question is whether we can obtain an approximate but simpler representation of at least some of these minima. This appendix provides an affirmative answer to that question for large values of the dimension N. We begin with the surface (22) which applies to the case of binary transmission:

$$\mathscr{F}^2 = 1 + \mathbf{c}\cdot\mathbf{c} - 2E|\mathbf{c}\cdot\mathbf{a}|. \tag{36}$$

The key is to note that if the vector $\mathbf{c}$ has many components approximately equal, then $\mathbf{c}\cdot\mathbf{a}$ will be approximately Guassian with mean zero and variance $\sum_1^N c_i^2$. Since, for a zero-mean Guassian variable having variance σ^2 we have $E|x| = \sqrt{2/\pi}\,\sigma$, (36) becomes

$$\mathscr{F}^2 = 1 + \sum_1^N c_i^2 - 2\sqrt{\frac{2}{\pi}}\sqrt{\sum_1^N c_i^2}. \tag{37}$$

$\mathscr{F}^2$ has a local minimum $\mathbf{c}$ is such that

$$\sqrt{\sum c_i^2} = \sqrt{\frac{2}{\pi}} = 0.798. \tag{38}$$

At the local minima we have

$$\mathscr{F}_0^2 = 0.363. \tag{39}$$

For four-level transmission the surface with which we must deal is described by (26). The presence of the function $\mathrm{sl}(\mathbf{c}\cdot\mathbf{a})$ only slightly complicates the calculations now; answers may readily be obtained numerically. We now have local minima whenever

$$\sqrt{\sum c_i^2} = 0.51, \tag{40}$$

and at the minima we have

$$\mathscr{F}_0^2 = 0.340. \tag{41}$$

Thus if N is large and $\mathbf{c}$ is not too close to any axis, we expect many minima located at the indicated radii, and all of about the same depth. Hence for these minima we expect the motion from one to the other to be more like free diffusion rather then leakage from a well. The difference in diffusion times for these two ideal situations is discussed in greater detail in Appendix II.

We close this appendix with a remark on the characteristic appearance of the equalizer output when its tap vector is trapped at a local minimum of the type just described (in contrast to the local minimum found at the end of Section III). The Gaussian assumption made concerning the distribution of $\mathbf{c}\cdot\mathbf{a}$, which is, in fact, the output, implies that the output will have a unimodal distribution, peaked at the origin,

and of variance indicated above. Such had been observed by Gitlin and Werner.[1]

APPENDIX B

Model Diffusion Problems

In discussing finite step-size effects in Section IV we suggested that, for small step size, a diffusion approximation would be a useful model for the random dynamics inherent in the adaptive algorithms that we are considering. We saw, further, that decision-directed startup procedures lead to a complicated region geometry for tbe error surface, and we mentioned that this precluded precise computation for the convergence rate of the optimum tap weights. However, an intuitive feeling for typical behavior certainly is worthwhile, and so we present in this appendix solutions to some simple but relevant model problems in diffusion.

A typical diffusion problem for our work would involve, say, finding the average time for a particle, starting at a given initial position, to diffuse to an error-free region. In setting up such a problem for solution, the boundary of the error-free region would be replaced by an absorbing barrier and the mean-first-passage time to hit the barrier would be required. Therefore, in our model problems, we treat situations where the starting point is surrounded by an absorbing barrier of simple form.

It is well known that one may approximate an isotropic random walk in N dimensions by a free diffusion.[4] If $p(\mathbf{x}, t; \mathbf{x}_0, t_0) \equiv p$ is the probability density for finding the particle at time t at position $\mathbf{x}$, given that at time t_0 it was as $\mathbf{x}_0$, then the density p obeys the diffusion equation[4]

$$\frac{\partial p}{\partial t} = D\nabla^2 p, \tag{42}$$

∇^2 being the N-dimensional Laplacian operator. The diffusion constant D is given by[4]

$$D = \frac{1}{2N} E \| \Delta \mathbf{x} \|^2 \rho, \tag{43}$$

where $\Delta\mathbf{x}$ is a step in the random walk that we are approximating by the diffusion, and ρ is the number of steps per unit time.

We have, of course, the initial condition for (42),

$$\lim_{t \to t_0} p = \delta(\mathbf{x} - \mathbf{x}_0). \tag{44}$$

Furthermore, there are the boundary conditions: $p = 0$ on an absorbing wall, while the normal derivative of p vanishes at a perfectly reflecting surface.

Our first task is to find an expression for the diffusion constant D in terms of the constants of the equalization problem. To this end, we rewrite (36) [restricting it to a given region and using (32)] as

$$\boldsymbol{\epsilon}_{n+1} = \boldsymbol{\epsilon}_n + \Delta\boldsymbol{\epsilon}_n - \alpha \mathbf{a}_n[\mathbf{a}_n \cdot \boldsymbol{\epsilon}_n], \tag{45}$$

with

$$\Delta\boldsymbol{\epsilon}_n = \alpha \mathbf{a}_n Q_0. \tag{46}$$

Equation (45) is then of the form of a random walk with a restoring term $-\alpha\mathbf{a}_n(\mathbf{a}_n \cdot \boldsymbol{\epsilon}_n)$. The quantity $\Delta\boldsymbol{\epsilon}_n$ alone, represents the steps that would be taken in a free-random walk, and thus $\Delta\boldsymbol{\epsilon}_n$ is to be identified with the step $\Delta\mathbf{x}$ in (43). Assuming for convenience an isotropic diffusion, we have, approximately,

$$E\,\|\Delta\boldsymbol{\epsilon}_n\|^2 \approx \alpha^2 (E\,\mathbf{a}_n^+ \mathbf{a}_n)(E\,Q_0^2) = N\alpha^2\sigma_a^2\mathscr{F}_0^2. \tag{47}$$

Thus if we identify the time t with n, so that ρ = one step/sec, we have, using (47) in (43),

$$D = \frac{\sigma_a^2 \alpha^2}{2}\mathscr{F}_0^2, \tag{48}$$

which is the expression for D that we seek.

To generalize (42) to include the effect of the restoring term in (45), we note that the diffusion equation may also be regarded as the Fokker–Planck equation[4] corresponding to the continuous time version of the random walk. The dynamical equation governing the latter would simply be

$$\frac{d\boldsymbol{\epsilon}}{dt} = \sqrt{2D}\,\mathbf{n}(t), \tag{49}$$

$\mathbf{n}(t)$ being a Gaussian white-noise vector, of zero mean, independent components, each component of which is normalized as

$$En(t)n(t') = \delta(t - t'). \tag{50}$$

Including the restoring term of (45) yields the following continuous-time dynamical equation approximating the motion:

$$\frac{d\boldsymbol{\epsilon}}{dt} = -\alpha\sigma_a^2\boldsymbol{\epsilon} + \sqrt{2D}\,\mathbf{n}(t), \tag{51}$$

where we have used (7) to obtain the first term of the right member.†

† The reader may wonder why no noise term appears in the dynamical equation analogous to the random dynamical term $\mathbf{a}_n\mathbf{a}_n^+\boldsymbol{\epsilon}_n$ of (30). The answer is simply that such a term is of higher order in α and we neglect it for simplicity. Theorems relevant to such small α diffusion approximations were first given by Kushner.[5] It was he who first suggested application of diffusion theory to stochastic approximation algorithms.

We simply state that the Fokker–Planck equation for the density $p \equiv p(x, t; \mathbf{x}_0, t_0)$ corresponding to this Markovian system is

$$\frac{\partial p}{\partial t} = \nabla \cdot [\alpha \sigma_a^2 \mathbf{x} p] + D \nabla^2 p. \tag{52}$$

The machinery just described is sufficient to solve some interesting problems. Obtaining the solutions for the simple problems that we consider is not difficult, and therefore only the results will be given. The detailed discussion up to this point was necessary for establishing the relationships between the constants appearing in our problem and those of diffusion theory.

Our first model problem is: What is the mean-first-passage time $\bar{t}$ for a particle to freely diffuse (no restoring force) to a surrounding sphere of radius R, in N dimensions?

The answer may be derived using the diffusion equation (42) and the average time turns out to be given by

$$\bar{t} = \frac{R^2}{2DN} = \frac{R^2}{\mathscr{F}_0^2} \frac{1}{\alpha^2 \sigma_a^2 N}.$$

This expression for the average first-passage time $\bar{t}$ implies that if, during decision-directed startup, we are in a region such as suggested in Appendix A and the step size α is, on the one hand, small enough for a diffusion approximation to hold, but yet is large enough so that small variations of $\mathscr{F}^2$ in going from one local minimum to a neighboring one are negligible, then we expect diffusion time out of the region to increase as $1/\alpha^2$. Further, if α is held at a fixed percentage of the typical value $\alpha = (1/N\sigma_a^2)$, then $\bar{t}$ is proportional to N, the number of equalizer taps.

We choose our second example to be one dimensional, for simplicity. A brownian particle starts at the minimum of a symmetric well, as shown in Fig. 5, and we assume that the points at $\pm R$ are absorbing. What is the average time $\bar{t}$ before the particle is absorbed, i.e., leaves the well?

If we take the equation of motion of the particle to be

$$\dot{x} = -kx + \sigma n(t), \tag{53}$$

where $n(t)$ is white noise as in (50), then, making appropriate use of the Fokker–Planck equation (52), we find

$$\bar{t} = \frac{R^2}{\sigma^2} \left[2 \int_0^1 \exp(-\theta v^2)\, dv \int_v^1 \exp(\theta w^2)\, dw \right] \equiv \frac{R^2}{\sigma^2} f(\theta), \tag{54}$$

where

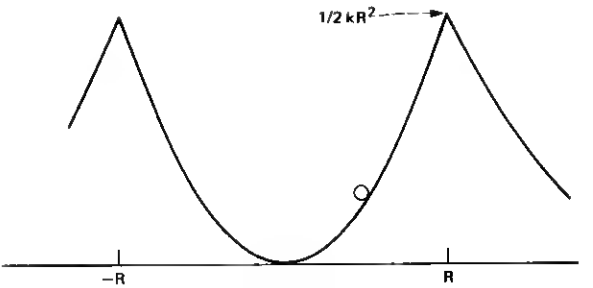

Fig. 5—Brownian particle in a well; $\dot{x} = -kx + \sigma n$.

$$\theta = k\frac{R^2}{\sigma^2}. \tag{55}$$

As for the properties of $f(\theta)$, we have $f(0) = 1$, $f(\theta) \geq 1$, and

$$f(\theta) \approx \begin{cases} 1 + \dfrac{\theta}{3}, & \theta \text{ small}, \quad (56a) \\ 1.44, & \theta = 1, \quad (56b) \\ \dfrac{e^{\theta}}{2\theta}\dfrac{\pi}{\theta}, & \theta \text{ large}. \quad (56c) \end{cases}$$

Using (48) and (51) to make contact with the equalization parameters, we have

$$\frac{R^2}{\sigma^2} = \frac{R^2}{2D} = \frac{R^2}{\mathscr{F}_0^2 \alpha^2 \sigma_a^2} \tag{57}$$

and

$$\theta = \frac{R^2}{\alpha \mathscr{F}_0^2}. \tag{58}$$

This last equation, in conjunction with (56c), shows that, as $\alpha \to 0$, the average trapping time for a particle in an isolated well grows exponentially with $1/\alpha$.

For the usual four-level decision-directed algorithm, we have already noted a local minimum at $\mathbf{c}_0 = (2/5, 0, 0, \cdots, 0)$; see Section IV and Fig. 4. Further, we saw that neither $\mathscr{F}_0^2$, nor the distance to the error free-region in the $\mathbf{c}_1$ direction (which distance we identify with the R of the above example), depended on the dimension N. However, by stability, α cannot exceed $2/N\sigma_a^2$. Therefore, since $\theta \approx (R^2/\mathscr{F}_0^2)\sigma_a^2 N$, it follows from (54) and (56c) that $\bar{t}$ would be enormously long for large N. In practice, for a 32-tap equalizer, the observed shrinkage of the output signal constellation appears to persist indefinitely. Numerically, from Fig. 3, we use $R = 0.667 - 0.400, \mathscr{F}_0^2 = 0.2$. Setting $\alpha = 1/N\sigma_a^2$, we

have that, for a 32-tap equalizer, $R^2/\sigma^2 = N\theta = 32\theta$, $\theta = 57$. Using (54) and (56c), this corresponds to 10^{25} iterations, on the average, before leaving the well.

REFERENCES

1. R. D. Gitlin and J. J. Werner, unpublished work.
2. E. Y. Ho, private communication.
3. S. Muroga, T. Tsuboi, and C. R. Baugh, "Enumeration of Threshold Functions of Eight Variables," IEEE Trans. Computers, *C-19* (Sept. 1970), pp. 818–825.
4. S. Chandrasekhar, "Stochastic Problems in Physics and Astronomy," reprinted in *Selected Papers on Noise and Stochastic Processes*, N. Wax, ed., New York: Dover, 1954.
5. H. J. Kushner and H. Huang, "Rates of Convergence for Stochastic Approximation Type Algorithms," to appear in Siam J. Control and Optimization.